
Article:
Timmis, Jonathan Ian orcid.org/0000-0003-1055-0471, Ismail, Amelia Ritahani Binti, Bjerknes, Jan D. et al. (1 more author) (2016) An Immune-Inspired Swarm Aggregation Algorithm for Self-Healing Swarm Robotic System. Biosystems. pp. 60-76. ISSN 0303-2647

https://doi.org/10.1016/j.biosystems.2016.04.001

An Immune-Inspired Swarm Aggregation Algorithm for Self-Healing Swarm Robotic Systems

Timmis, J.*

Department of Electronics, University of York, Heslington, York. YO10 5DD

Ismail, A. R.

Department of Computer Science,
Kulliyyah of ICT, International Islamic University Malaysia,
P.O. Box 10, 50728, Kuala Lumpur, Malaysia

Bjerknes, J.D.

Kongsberg Defence Systems, P.O. Box 1003, NO-3601, Kongsberg, Norway.

Winfield, A.F.T

Faculty of Environment and Technology, University of the West of England, Bristol BS16 1QY, UK

1. Introduction

Swarm robotics is an approach to the co-ordination and organisation of multi-robot systems of relatively simple robots [13]. When compared to traditional multi-robot systems that employ centralised or hierarchical control and communication systems in order to coordinate behaviours of the robots, swarm robotics adopts a decentralised approach in which the desired collective behaviours emerge from the local interactions and communications between robots and their environment. Such swarm robotic systems may demonstrate three desired characteristics for multi-robot systems: *robustness, flexibility and scalability.* [1] define these characteristics as:

- robustness is the degree to which a system can still function in the presence of partial failures or other abnormal conditions;
- flexibility is the capability to adapt to new, diverse, or changing requirements of the environment;

*Corresponding author

Email addresses: jon.timmis@york.ac.uk (Timmis, J.), amelia@iium.edu.my (Ismail, A. R.), jan.dyre.bjerknes@kongsberg.com (Bjerknes, J.D.), alan.winfield@uwe.ac.uk (Winfield, A.F.T)

- scalability can be defined as the ability to expand a self-organised mechanism to support larger or smaller numbers of individuals without impacting performance considerably

Even though the swarms exhibit high level of robustness, such claims are frequently not supported by empirical or theoretical analysis [2]. Those authors also explored fault tolerance in robot swarms through Failure Mode and Effect Analysis (FMEA) illustrating by a case study of wireless connected robot swarm, in both simulation and real laboratory experiments. A failure mode and effects analysis (FMEA) is a procedure for analysis of potential failure modes within a system for classification by severity or determination of the effect of failures on the system. Failure modes are any errors or defects in a process, design, or item, leading to the studying the consequences of those failures to the systems. The FMEA case study in [2] showed that a robot swarm is remarkably tolerant to the complete failure of robot(s) but is less tolerant to partially failed robots. For example, a robot with failed motors but with all other sub-systems functioning, can have the effect of anchoring the swarm and hindering or preventing swarm motion (taxis toward the target). [2] then concluded that: (1) analysis of fault tolerance in swarms critically needs to consider the consequence of partial robot failures, and (2) future safety-critical swarms would need designed-in measures to counter the effect of such partial failures. One of the example is to envisage (form) a new robot behaviour that identifies neighbours who have partial failure, then 'isolates' those robots from the rest of the swarm: a kind of built-in immune response to failed robots [2].

Work in [13] argues that a significant benefit of swarm robotics is robustness to failure. However, recent work has shown that swarm robotic systems are not as robust as first thought [3, 2]. This is also described in [12, 6], which highlighted that the number of foods collected in the foraging environment decreased when the stopped or failed robots increases in a simulation field with hundreds of swarm robots.

To demonstrate the above-mentioned robustness issue, a simple but effective algorithm for emergent swarm taxis (swarm motion towards a beacon) is proposed by [3]., In order to achieve beacon-taxis, the algorithm allows the swarm to move together towards an IR beacon using a simple symmetry breaking mechanism without communication between robots. To aid understanding of the reliability of the swarm, the evaluation of the effect of individual robot failure towards the operation of the overall swarm were investigated, these being: (1) complete failures of individual robots due to a power failure (2) failure of a robot's IR sensor and (3) failures of robot's motors leaving all other functions operational including the sensing and signalling. The study revealed that the effect of motor failures will have a potentially serious effect of causing the partially-failed robot to 'anchor' the swarm impeding the movement towards the beacon.

Work proposed in this paper attempts to address the issue of the emergence of 'anchor points' under the case of partial failure of robots as outlined above. We propose an immune-inspired approach which enables, under certain failure

modes, the swarm to 'self-heal' and continue operation and continue operation and demonstrate emergent swarm taxis. We propose an extension to the existing ω-algorithm [3] that affords a self-healing property that functions under certain failure modes. Our solution takes inspiration from the process of granuloma formation, a process of 'containment and repair' observed in the immune system. We then derive a set of design principles that we use to instantiate an algorithm capable of isolating the effect of the failure, initiate a repair sequence to allow the swarm to continue operation. This idea was first discussed in [7] and in further detail by [20]. For the purposes of this paper, our scenario is that of power failure in a robot which results in the loss of mobility for the robotic unit, but limited communication is still possible. We explore the performance of the proposed algorithm in simulation and show that by having this new 'granuloma formation algorithm', effective energy sharing can be initiated between the failed robot(s) that allow for a repair to be made that allows for aggregation of the swarm to continue.

This paper is structured as follows. Section 2 reviews the swarm taxis algorithm that is used as the case study in this paper, and acts as the baseline performance for our experimental work. We then present our experimental design in Section 3 and review specific issues that we intend to resolve with the immune inspired approach. In Section 4, we outline the concept of granuloma formation from an immunological perspective. This is then followed by a proposed granuloma formation algorithm that has been developed from the immunology using modelling and derivation of design principles. We finally provide conclusions and discussions on our future work in Section 5.

2. Swarm Taxis Algorithm

Aggregation of a swarm requires that agents have a physical coherence when performing a task. Robots are randomly placed in an environment and are required to interact with each other to maintain physical coherence and complete a task. This is relatively easy when a centralised control approach is used but very challenging when a distributed control approach is used [1]. [11], [3] which was published as [2] developed a class of aggregation algorithms which makes use of local wireless connectivity information alone to achieve swarm aggregation namely the α, β and ω algorithms.

The α, β algorithms used "situated communications" [17], in which connectivity information is linked to robot motion so that robots within the swarm are wirelessly 'glued' together. This algorithm is inspired by the framework of minimalist design introduced by [8], which focused on very limited robots that are able to communicate locally but lack global knowledge of the environment. The only sensor information available, besides the basic obstacle avoiding infra-red (IR)sensors, are the beacon sensors and the radio communication. It is assumed that the communication hardware has a limited range, is omni-directional and the quality of the transmission is not optimal. The aim is to keep the robot as simple as possible as it is believed that stability is reachable only with a limited range radio device and proximity sensors for avoidance. The key idea is that

the limited range gives enough information on relative position. Such severely constrained conditions allow the act of communication with other behaviours of the robot and referred as situated communication in[16].

For our experimental case study we make use of a ω-algorithm with two swarm behaviours: flocking and swarm taxis toward a beacon. The combination means that the swarm maintains itself as a single coherent group while moving toward an infra-red (IR) beacon. The algorithm is a modified version of the wireless connected swarming algorithm (the α-algorithm) developed by Nembrini et al [11]. The wireless communication channel in ω-algorithm has been removed and replaced with simple sensors and a timing mechanism. Flocking is achieved through a combination of attraction and repulsion mechanisms. Repulsion between robots is achieved using IR sensors and a simple obstacle avoidance behaviour. Attraction is achieved using a simple timing mechanism. Each robot measures the duration since its last avoidance behaviour, and if that time exceeds a threshold, then the robot turns towards its own estimate of where the center of the swarm is and moves in that direction for a specified amount of time. In order to reach the beacon, the algorithm uses a symmetry-breaking mechanism, in which the short-range avoid sensor radius for those robots that are illuminated by the beacon is set slightly larger than the avoid sensor radius for those robots in the shadow of other robots [3]. An emergent property of this approach is swarm taxis towards the beacon.

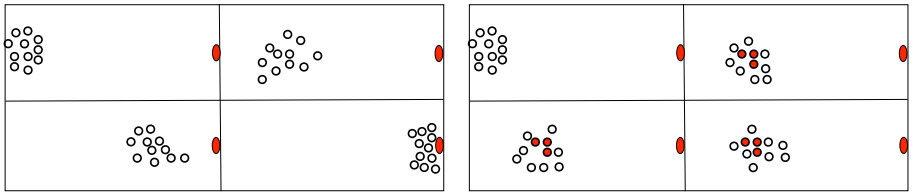
2.1. Anchoring of the Swarm

Various types of failure modes and the effect of individual robots failures and its effect to the swarm have been analysed by [2]. Quoted from [2] the failure modes and effects for swarm beacon taxis are as follows :

- Case 1: complete failures of individual robots (completely failed robots due, for instance, to a power failure) might have the effect of slowing down the swarm taxis toward the beacon. These are relatively benign, in the sense that 'dead' robots simply become obstacles in the environment to be avoided by the other robots of the swarm.
- Case 2: failure of a robot's IR sensors. This could conceivably result in the robot leaving the swarm and becoming lost. Such a robot would become a moving obstacle to the rest of the swarm and might reduce the number of robots available for team work.
- Case 3: failure of a robot's motors only. Motor failure only leaving all other functions operational, including IR sensing and signalling, will have the potentially serious effect of causing the partially-failed robot to 'anchor' the swarm, impeding its taxis toward the beacon.

The effect of partially-failed robot to 'anchor' the swarm leading to impeding its taxis toward the beacon is shown pictorially in Figure 1. This is the effect of the failure of robots motors. If in the swarm there is only one or two robots

that fail, the swarm can still moves towards the beacon. This is a form of *self-repairing* mechanisms inherent in the ω-algorithm. For example, with complete failure of a robot, the 'dead' robots simply become obstacles in the environment to be avoided by the other robots of the swarm. However the swarm will also experience a serious effect which cause the partially failed robot to 'anchor' the swarm. In swarm beacon taxis, this can only happen if the *anchoring force* resulted by the effect of the robot's motor are greater than the *beacon force*, which is the force from the beacon in the systems.



(a) Swarm beacon taxis without failure. Here we see the swarm successfully moves over to the beacon on the right-hand side of the image

(b) Swarm beacon taxis where 3 robots fail simultaneously. Here we see the swarm become stationery, and an anchor point emerge, with the swarm now being unable to continue towards the beacon as described in case 1 and case 3

Figure 1: Swarm Beacon Taxis with and without anchoring

In emergent swarm taxis, a certain number of robots are necessary in maintaining the emergence property. A reliability model (k-out-of-n-system model) of the swarm in swarm beacon taxis has been developed and the results show that there is a point at which swarms no longer are as reliable as first thought [2]. The result suggests that in a swarm of 10, then at least 5 robots have to be working in order for swarm taxis to emerge. This would indicate that in order for the swarm to continue operation then some form of *self-healing* mechanism is required apart from *self-repairing* mechanism which is already available in swarm beacon taxis [3].

Further work in this paper, will consider the failure mode of the failed motor, with the addition of the cause of motor failure is lack of power, but with the assumption enough power remains for simple signalling, but not enough to power the motors. This assumption has been tested electronically using e-puck robots with a simple obstacle avoidance task. We performed a simple experiment with epuck robots where we allowed the robots to wander in an environment with a simple obstacle avoidance behaviour until the robots stopped moving. We monitored the power levels within the robots form this point (when the robots stopped moving) and when the battery was totally discharged and the result is shown in table 1. We found that on average, the e-puck robots are able to send signals for 27 minutes before all the energy is lost. This failure mode allows us to construct a potential self-healing mechanism for the swarm to, which involves

a recharge of drained batteries.

Table 1: Results for robot's wheel and robot's led

Robots No	Difference in time between the robot's wheels stop moving and robot's led stop signalling
Robot 1	27 minutes 36 seconds
Robot 2	29 minutes 37 seconds
Robot 3	28 minutes 29 seconds
Robot 4	25 minutes 48 seconds

The following section investigate ω-algorithm [3] in Player/Stage simulation that will serve as a baseline from which we will calibrate our new proposed system against. In addition, we propose a series of potential solutions to the anchoring issue, and evaluate their performance when compared to each other. What we find is that no approach is suitable, and this leads to the development of a more sophisticated extension of the ω-algorithm inspired by the immune system.

3. Initial Investigations into the ω-algorithm

3.1. Experimental Protocol

The experiments presented in this section were performed in simulation using the sensor-based simulation tool set, Player/Stage [5][1]. 10 e-puck robots are simulated, sized 5cm x 5cm, and equipped with 8 proximity sensors, two at the front, two at the rear, two at left and two at right. Initially robots are randomly dispersed within a 2 m circle area with random headings. A robot will poll its proximity sensors at frequency 5 Hz (1/T), whenever one or more sensors are triggered the robot will execute an avoidance behaviour, and turn away from the colliding robot or obstacles. The avoidance turn speed depends on which sensors are triggered and robot will keep turning for 1 second. The task of the swarm is to aggregate and move together towards an infra-red beacon. The environment is a 20 x 20 cm square arena with a beacon at position (4.0, 0.0) as shown in figure 2. The fixed parameters for the simulation are displayed in Table 2. Each simulation run consisted of ten robots and was repeated ten times. For each run, the centroid position of the robots in swarm were recorded. For the first experiment, we developed ω-algorithm [3] in simulation and serve as our baseline behaviour for the remaining experiments.

As outlined above, the failure mode for our experiments is a motor failure, which is a consequence of a partial failure in the power unit. We assume that the failure is sufficient to stop the motors of the robot, but that there is sufficient power to light LEDs. Using the simulation, we inject a power reduction failure

[1]Player-Stage can be downloaded from http://playerstage.sourceforge.net/

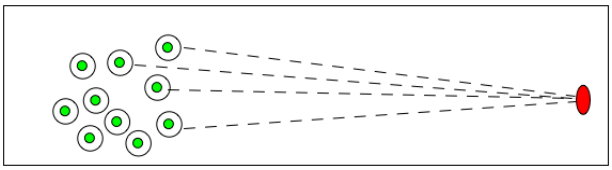


Figure 2: Snapshots of the simulation with Player/Stage with 10 robots and 1 beacon

Table 2: Robot fixed parameters for all simulations.

Parameter	Value
Time step duration	1 s
Robot normal speed	0.15cm/s
Avoidance sensor range	0.4cm
Robot body radius	0.12cm
Wireless range	2.0cm
Minimum energy needed to move	500j
Battery capacity	5000j
Component fault	power drain
No of faulty robots	1 to 5 units
Simulation duration	1000 seconds

to robots: for a single robot failure we term it R_{SF}, for two robot failures R_{TF} and three or more robot failures (until 5) we term them R_{TM}. In R_{SF}, one robot in the environment will experience a power reduction approximately at time=100 seconds in the simulation. For R_{TF}, two robots will be supplied with a power reduction simultaneously at t=100 seconds whilst with R_{MF}, three and more failing robots will be introduced simultaneously in the simulation. During this time, the robots are not moving and will remain static in the environment. The parameter for the faulty scenario is shown in Table 3.

Table 3: Variable parameters for failing scenario in the environment

Number of faults	Parameter	Time (s)
Single failure	Speed=0 m/s, energy=500 joules	t=[100]
Multiple failure	Speed=0 m/s, energy=500 joules	t=[100]

In these experiments we measure the progression of the centroid of the swarm towards the beacon for every 100 seconds, using equation 1; where x and y are the coordinates of the robots and n is the number of robots in the experiment.

$$\text{Centroid distance of robots to beacon} = \sum_{i=1}^{n} \frac{\sqrt{(x_{1_i} - x_{2_i})^2 + (y_{1_i} - y_{2_i})^2}}{n} \quad (1)$$

3.2. Results and Analysis

3.2.1. Experiment I: ω-algorithm

$H1_0$: *The use of ω-algorithm (M1) for swarm beacon taxis allows the swarm to achieve a centroid distance less than 5 cm away from the beacon when there are no failures introduced.*

$H2_0$: *The use of ω-algorithm (M1) for swarm beacon taxis allows all robots in the swarm to achieve the distance less than 5 cm away from the beacon with a completely failed robot in the environment.*

$H3_0$: *The use of ω-algorithm (M1) for swarm beacon taxis allows all robots in the swarm to achieve the distance less than 5 cm away from the beacon with more than two failing robots in the environment.*

Our experiment starts with the investigation of ω-algorithm for swarm beacon taxis, in effect $H1_0$ as undertaken by [3]. The swarm starts in one part of the arena and moves toward the beacon. The distance from the centroid of the swarm to the beacon for each run is given in Figure 3. For each experiment the robots have a different starting position in the arena, but as the importance here is on the relative performance between different sets of runs, the starting point was set to 35 cm from the beacon. This allows for a comparison between each run, as comparison between runs will be for identical starting distances from the beacon. The hypothesis can be accepted if the swarm centroid comes close to the beacon within a range of less than 5 cm. Based on the experiments, the swarm has a mean of velocity of 1.2 cm/simulation seconds. The fastest in the experiment of 10 robots moved at 1.52 cm/simulation seconds and the slowest had the velocity of 1.01 cm/simulation seconds. At time t=600 seconds, swarm has reached 5 cm away from the beacon. We then use the sample data to construct a 95% confidence interval for the mean centroid position of the robots from the beacon (assumed normal) and we obtained 95% confidence interval in our experiment.

In the context of the scenario above, we then introduced a partial failure to an individual robot in the swarm, to test $H2_0$. The swarm will experience a temporary slow down as it is attempting to avoid the obstructing robots, then it will pick up its normal velocity once the failed robot has been avoided. The mean distance over time in the ten runs is shown in figure 4. The swarm has the mean velocity of 1.32 *cm/sec*, where the fastest velocity in the experiment is 1.65 *cm/sec* when the swarm is trying to free itself from the failed robot. During the faulty scenario, the robot does not move and remains static. The experiment accept the hypothesis $H2_0$ at the default of α= 0.05 significance level, which is indicated by the p value = 0.6442 is much greater than the α. The 95% confidence interval on the mean centroid distance of robots to the beacon is less than 5 *cm* is obtained from this experiment.

In the third set of experiments with two and then three completely failing robots are introduced to the simulation. This experiment will test $H3_0$. Figure

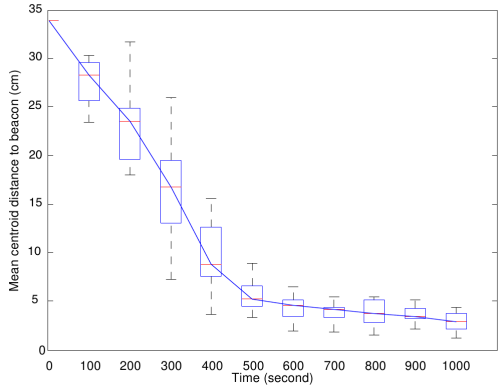


Figure 3: Boxplots of the distance between swarm centroid and beacon as a function of time for 10 experiments using ω-algorithm with no robot failure for $H1_0$. The centre line of the box is the median while the upper edge of the box is the 3^{rd} quartile and the lower edge of the box is the 1^{st} quartile. At about t=600 seconds, the swarm has reached the beacon

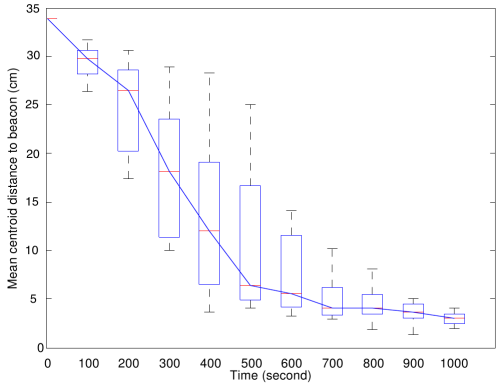
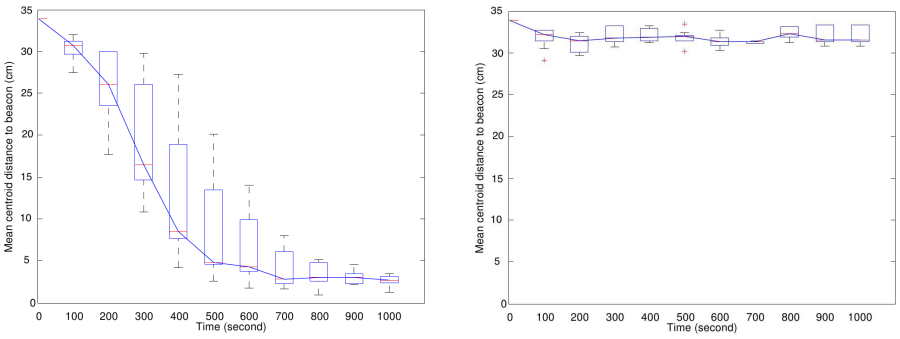


Figure 4: Boxplots of the distance between swarm centroid and beacon as a function of time for 10 experiments using ω-algorithm with one robot fails at t=100 seconds for $H2_0$. The centre line of the box is the median while the upper edge of the box is the 3^{rd} quartile and the lower edge of the box is the 1^{st} quartile. The swarm reach the beacon at t=700 seconds

5(a) and 5(b) show the results of multiple robots failure in swarm beacon taxis. As mentioned by [3], the influence from this two failed robots are small and the failed robots again will be avoided as if they were obstacles in the environment, and the swarm will continue towards the beacon. This is confirmed in figure 5(a). However, as more failed robots are introduced in the experiment, the swarm starts to stop moving towards the beacon rather stagnate around the failed robots that is now act as the 'anchor' and they will never reach the beacon as shown in figure 5(b). Form these experiments, even though the robots are

able to reach less than 5 *cm* away from the beacon when two failing robots are introduced to the system, the anchoring problem will manifest as three failing robots are introduced to the systems. With three failed robots, the experiment reject the hypothesis $H3_0$ at the default of α= 0.05 significance level, which is indicated by the p value = 3.6926e-14 that has fallen below the α. The 95% confidence interval on the mean centroid distance of robots to the beacon is more than 5 *cm* is obtained from this experiment.



(a) Boxplots of the distance between swarm centroid and beacon as a function of time for 10 experiments using ω-algorithm with two robot fails simultenously at t=100 seconds for $H2_0$. The centre line of the box is the median while the upper edge of the box is the 3^{rd} quartile and the lower edge of the box is the 1^{st}quartile. The swarm reach the beacon at t=850 seconds

(b) Boxplots of the distance between swarm centroid and beacon as a function of time for 10 experiments using ω-algorithm with three robot fails simultaneously at t=100 seconds $H3_0$. The centre line of the box is the median while the upper edge of the box is the 3^{rd} quartile and the lower edge of the box is the 1^{st} quartile. The swarm does not reach the beacon as the failing robots anchor the swarm

Figure 5: The ω-algorithm with two and three failing robots

Results from the experiments show that, even with two completely failed robots the swarm will always reach the beacon, and the delay is once again relatively small. However, as three faulty robots are introduced in the simulation, the swarm starts to show a complete halt, not reaching the beacon and stagnate around the three failing robots as shown in figure 5(b). This confirms the potential issue of "anchoring" in swarm beacon taxis that is described in section 2.1. From all experiments that have been conducted, we have confirmed the observations of [3] that the anchoring issue, occur in the swarm beacon taxis as more failures are injected in the systems. The failing robot will 'anchor' the swarm impeding its movement towards the beacon.

Now we have reproduced the effect reported by [3] we propose a number of solutions that might potentially mitigate the effect of the observed anchoring issues. We call these approaches the single nearest charger and shared nearest

charger algorithms.

3.2.2. Experiment II: Single Nearest Charger Algorithm

$H4_0$: *The use of single nearest charger mechanism (M2) for swarm beacon taxis does not improve the ability of the robots in the systems to reach the beacon when compared with M1 when more than two faulty robots are introduced to the systems*

As discussed by [9], in contrast of the form of 'central sharing', robots must be able to distribute the collective energy resources owned by the group member (trophallaxis). Taking this idea, we apply a simple solution in energy sharing between robots in the simulation called the single nearest charger algorithm. We begin by assuming the simplest rules basically depend on the each robot's position in the environment. We limit the energy transfer between two robots, meaning that each robot can only receive or donate energy from one robot at any one time [9]. We also constrain the system such that the nearest robot that acts as the donor cannot reject the request and must donate the amount of energy defined by the energy threshold in the simulation. In our experiment, we set the energy threshold = 1500 Joules, thus each donor can only transfer the up to the energy threshold to preserve the donor's energy. This algorithm is outlined in algorithm 1.

This experiment tests hypothesis $H4_0$ to determine whether better results can be obtained from a single charger algorithm that is described in algorithm 1. Thus, the first experiment tests whether the single charger algorithm is able to allow the systems to move at least 5 *cm* away from the beacon with failing robots in the environment.

begin
  Deployment: robots are deployed in the environment
  repeat
    Random movement of the robot in the environment
    Signal propagation: Faulty robots will emit faulty signal
    Rescue: One of the healthy robots with the nearest distance (earliest arriving robot) will perform protection and rescue
    Repair: Sharing of energy between faulty and healthy robots according to algorithm 2
  until *forever*
end

Algorithm 1: Overview of single nearest charger algorithm

Algorithm 1 reflects the overview of single nearest charger algorithm to do the repair for the faulty robot(s). The basic terms used in the algorithm are outlined below:

- $pos_{self}(t)$: position of the current robot
- $pos_{peer}(t-x)$: position of peer robots

begin
Evaluate $pos_{self}(t)$
Send $pos_{self}(t)$ *to peers*
Receive $egy_{peer}(t-x)$ *and* $pos_{self}(t-x)$ *from peers*
forall $egy_{peer}(t-x)$ **do**
Evaluate $egy_{peer}(t-x)$
if $egy_{peer}(t-x) < egy_{min}$ **then**
Evaluate $pos_{peer}(t-x)$
Sort $pos_{peer}(t-x)$*in ascending order*
Move to nearest $pos_{peer}(t-x)$
else
Do not move to $pos_{peer}(t-x)$
end
end
end

Algorithm 2: Algorithm for containment and repair for single nearest charger algorithm

- $egy_{self}(t)$: energy of the current robot
- $egy_{peer}(t-x)$: energy of peer robots
- egy_{min}: minimum energy required
- egy_{needed}: energy needed by each robot

Similar to the previous set of experiments in M1 (our baseline), M2 is tested with the same scenarios and results are compared to M1. We look at both statistical and scientific significance. Statistical significance is measured with ranksum test and scientific significance with A measure [21]. The interpretation of A value is such that a value around 0.5=*no effect*, 0.56=*small effect*, 0.64=*medium effect*, and 0.71=*big effect*. A p value of 0.05 for ranksum test is commonly used to signify that two samples are different with the statistical significance because the medians are different at a 95% confidence level.

In the first experiment with R_{SF} and R_{TF}, M2 does not differ greatly from M1 which is indicated by the p value=0.54 with one robot fails and p value=0.8150 with two robots fail. From this results the swarm can get within 5 *cm* of the beacon. However, as compared to M1, in M2 all robots are able to arrive to the beacon whereby in M1, even though with one and two failing robots in the swarm, the swarm can reach the beacon but they leave the failing robots behind and avoid them as obstacles: this is consistent with the observations by [3] and [2] as discussed in Section 2.1. The results for M2 for one and two failing robots are shown in figure 6(a) and 6(b).

With three and four failed robots in the system, the experiment rejects the hypothesis $H4_0$ at the default of α= 0.05 significance level, which is indicated by the p value = 1.7661e-04 and p value = 1.7562e-04 that have fallen below the α. Thus, M2 performs significantly better from M1 when there are three

and four failed robots in the system. Results for three and four failing robots are shown in figure 6(c) and figure 6(d). For five failed robots M2 does not gives a better performance if compared with M1 as indicated by the p value = 0.3033. The difference between M2 and M1 is shown in table 4. Therefore, with three and four failed robots, the experiments reject the hypothesis $H4_0$ with 95% confidence level on the mean centroid distance of the robots to the beacon is more than 5 *cm* is obtained.

Table 4: P and A value for mean centroid distance on robots between M1 and M2 in 1000 simulation seconds.

M2/M1	**ranksum P**	**A measure**
One fail	0.54	0.7908
Two fails	0.8150	0.0181
Three fails	1.7661e-04	1
Four fails	1.7562e-04	1
Five fails	0.3033	0.640

The robots start with same levels of energy which is 5000 Joules. In each failing case, the energy is reduced as explained in table 3. The threshold value for the failing robots to start triggering failing signal is 500 J. Figure 7(a), 7(b) and 7(c) show the energy of each robot during 1000 simulation seconds with different failing robots in the environment. In figure7(b), three failing robots are introduced into the environment. Half of the swarm reach the minimum energy threshold in 900 simulation. From figure 7(c) we can see that for four failing robots, the swarm starts to reach the minimum energy threshold during 900 simulation seconds. This trend continues for five failing robots in the environment.

In summary, results from this experiment show that M2 performs slightly better than M1 but the swarm cannot reach the beacon if the number of failing robots is more than three. During the experiments, we also observe that the nearest charging robot may not have sufficient energy to donate to the faulty robots leading to the failure of the charging robots. This method would also be applicable if the number of failing unit is small (less than half of the swarm fails). However as more robots fail, they will then act as 'anchor' the swarm leading to impeding the swarm to moves towards the beacon. Although M2 managed to achieve desirable solution with R_{SF} and R_{TF} but, for R_{MF}, is unable to reach the beacon in all simulated scenarios. Thus, the next set of experiments in the following section investigates an enhanced method and introduces the shared nearest charger mechanism.

3.2.3. Experiment III: Shared Nearest Charger Algorithm

$H5_0$: *The use of a shared nearest charging mechanism (M3) does not improve the ability of the robots in the systems to reach the beacon when compared to M1 and M2 when more than two faulty robots are introduced to the systems*

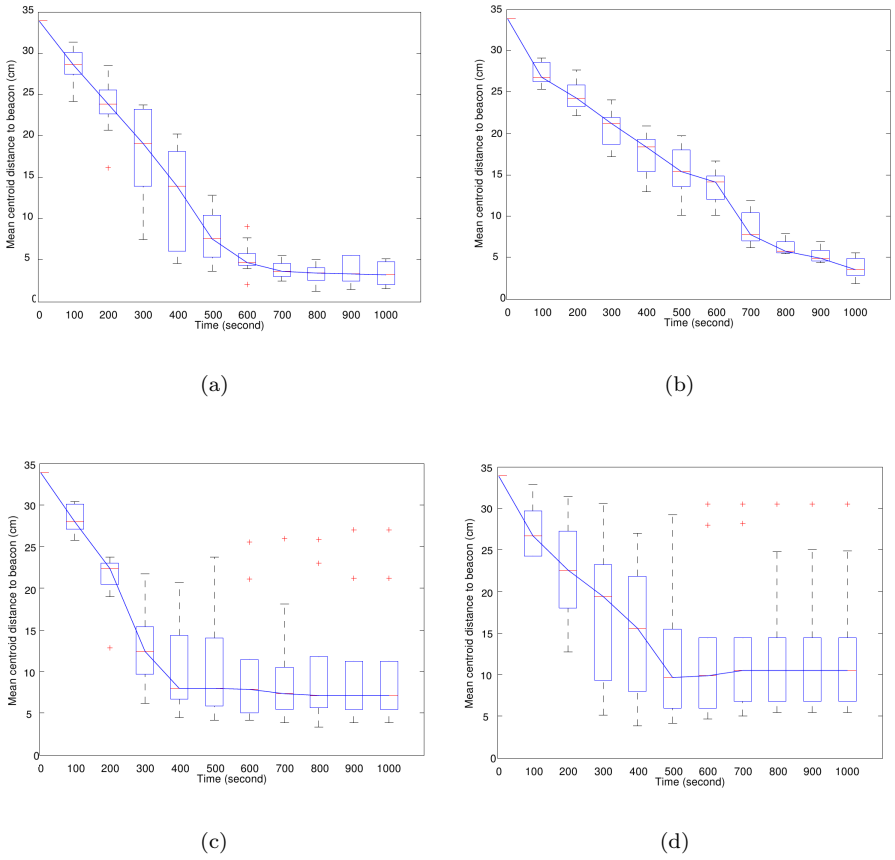
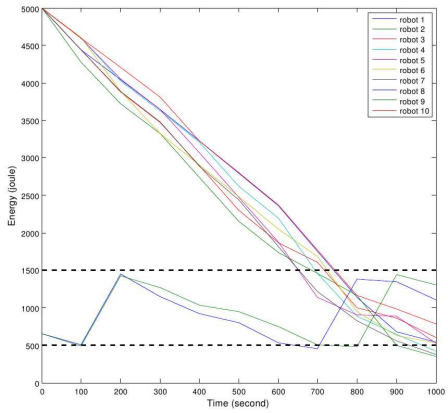


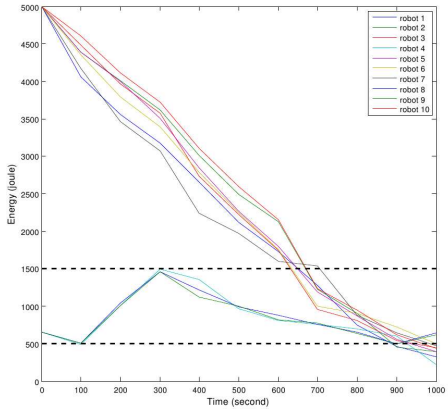
Figure 6: Boxplots of the distance between swarm centroid and beacon as a function of time for 10 experiments using single nearest charger algorithm with one faulty robot (6(a)), two robots fail simultenously at t=100 seconds(6(b)), three robots fail simultenously at t=100 seconds (6(c)) and four robots fail simultenously at t=100 seconds (6(d)). The centre line of the box is the median while the upper edge of the box is the 3^{rd} quartile and the lower edge of the box is the 1^{st} quartile. In figure 6(a), with one faulty robot, the swarm reach the beacon at t=650 seconds. With two failing robots in figure 6(b), the swarm reach the beacon at t=900 seconds. Meanwhile as depicted in figure 6(c) and figure 6(d) the swarm does not reach the beacon as the failing robots 'anchor' the swarm to moves towards the beacon.

Here we investigate whether M3 with a shared charger mechanism can outperform M2 and M1 with a mean centroid distance of robots from the beacon is 5 cm . Based on the idea of single nearest charging, we extend the algorithm by increasing the number of donors to each faulty robot. The general algorithm of shared nearest neighbour algorithm is presented in algorithm 3, which is also illustrated in figure 8.

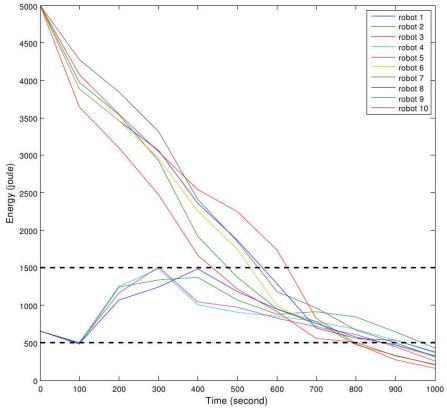
From figure 8, it is shown that three donor robots exist to share their energy



(a)



(b)



(c)

Figure 7: Figure 7(a) shows the energy of ten robots with two robots fail, figure 7(b) shows the energy of ten robots with three robots fail using single nearest charger algorithm and figure 7(c) shows the energy of ten robots with four robots fail. When two robots fail, as illustrated in figure 7(a), the energy of half of the swarm has reached the minimum energy threshold during 900 simulation seconds. With the case of three and four failing robots, the energy of the swarm has reached the minimum energy threshold during 900 simulation seconds as shown in figure 7(b) and figure 7(c).

begin
Deployment: robots are deployed in the environment
repeat
Random movement of the robot in the environment
Signal propagation: Faulty robots will emit faulty signal
Rescue: n number healthy robots with the nearest distance (earliest arriving robot) and highest energy will perform protection and rescue according to algorithm 4
Repair: Sharing of energy between faulty and healthy robots according to algorithm 2
until *forever*
end

Algorithm 3: Overview of Shared Nearest Charger Algorithm

with the failed robot. In this algorithm, the robots can transfer the minimum amount of energy from each of the neighbouring/charging robots taking into account that priority will be given to the robots with higher energy and near to the failing robots. In the algorithm, we limit the energy transfer between three robots, which means that each faulty robot can only receive energy simultaneous from three nearest robots. The donor must donate the amount of energy defined by the energy transfer rule in algorithm 4.

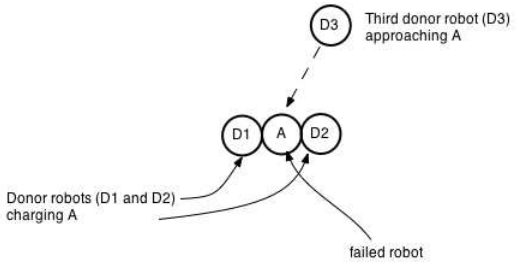


Figure 8: The illustration of shared nearest charger algorithm

The basic terms used in the algorithm are as follow:

- $pos_{self}(t)$: position of current robot
- $pos_{peer}(t-x)$: position of peer robots
- $egy_{self}(t)$: energy of current robot
- $egy_{peer}(t-x)$: energy of peer robots
- egy_{min}: minimum energy required
- egy_{needed}: energy needed by each robot

```
begin
    Evaluate pos_self(t)
    Send pos_self(t) to peers
    Receive egy_peer(t − x)/ and pos_self(t − x) from peers
    forall egy_peer(t − x) do
        Evaluate egy_needed(t)
        Divide egy_needed(t) with n
        Send egy_needed(t) to peers
        if egy_peer(t − x) < egy_min then
            Evaluate pos_peer(t − x)
            Sort pos_peer(t − x)in ascending order
            Move to nearest pos_peer(t − x)
        else
            Do not move to pos_peer(t − x)
        end
    end
end
```

Algorithm 4: Algorithm for containment and repair according to energy and position of robots

- n: number of donor robot, in this experiment n = 3

As with previous experiments in section 3.2.1 and section 3.2.2, M3 is tested in the same scenarios and results compared to M2 and M1. We again look at both statistical and scientific significance.

In the first experiment with R_{SF} and R_{TF}, results of M3 do not differ greatly from M1 as in both methods, the swarm cannot reach the beacon if the number of failing robots is more than three. However, as compared to M1, in M2 all robots are able to arrive at the beacon. The results for M3 are shown in figure 9(a), 9(b), 9(c) and 9(d). Based on figure 9(a) and figure 9(b), M3 works well with one and two failing robots in the systems where the swarm can reach the beacon. However, it starts to suffer when the number of failures increases. With three, four and five failing robots in the system, the swarm fails to reach the beacon. These are shown in figure 9(c) and figure 9(d). In the experiment with R_{SF} and R_{TF}, results of M3 do not differ much from M1 and M2 as in all methods, all robots in the swarm can reach the beacon. These are illustrated in figure 9(a) and figure 9(b). However, this algorithm is unable to perform the task in reaching the beacon with R_{MF}. As depicted in figure 9(c) and figure 9(d), they can only reach around 10 to 15 *cm* away from the beacon. We will later describe the significance differences between M3, M2 and M1.

Both algorithms do not differ statistically with R_{SF} and R_{TF} but, for R_{MF}, M3 performs significantly better than M2 which is indicated by the p value = 0.4261, p value = 0.0930 and p value = 0.8785. The difference is statistically and scientifically significant, as shown in Table 5. However, as the number of failures reaches five (half of the swarm) both methods show no difference in performance. Based on the experiments conducted, when half of the swarm

fails the remaining robots are not able to reach the beacon. This scenario is observed during simulation M1 and M2 for swarm beacon taxis, and again is consistent with similar observations by [2].

Table 5: P and A value for mean centroid distance on robots between M3 and M2 in 1000 simulation seconds.

M3/M2	**ranksum P**	**A measure**
One fail	0.3631	0.3750
Two fails	0.2404	0.34
Three fails	0.4261	0.6100
Four fails	0.0930	0.7200
Five fails	0.8785	0.4750

M3 and M1, do not differ statistically with R_{SF} and R_{TF} but, with three and four failing robots in the system, the experiment reject the hypothesis $H5_0$ at the default of α= 0.05 significance level, which is indicated by the p value = 1.7462e-04 and p value = 1.0650e-04 that have fallen below the α. Thus, M3 performs significantly better from M1 when there are three and four failing robots in the system. For five failing robots M3 does not gives a better performance if compared with M1 as indicated by the p value = 0.4226. The difference between M3 and M1 is shown in table 6. Therefore, with three and four failing robots, the experiments reject the hypothesis $H4_0$ with 95% confidence level on the mean centroid distance of the robots to the beacon is more than 5 cm .

Table 6: P and A value for mean centroid distance on robots between M3 and M1 in 1000 simulation seconds.

M3/M1	**ranksum P**	**A measure**
One fail	0.650	0.2718
Two fails	0.57	0.62
Three fails	1.7462e-04	1
Four fails	1.0650e-04	1
Five fails	0.4226	0.61

As in previous experiments, all robots start with equal energy which is 5000 Joules. Figure 10(a), 10(b) and 10(c) explains the energy of 10 robots during 1000 simulation seconds with different failing robots in the environment with shared charging algorithm. For two and three robot failures, the robots do not suffer with minimum energy level. The charging robots have enough energy to recharge the faulty robots and maintain a significant amount of energy to move towards beacon. These are shown in figure 10(a) and figure 10(b). However, for four and five failing robots the results differ from the two and three failing robot cases. The robots start to lose a significant amount of energy and towards the end of the simulation, half of the swarm has energy less than the threshold value. Furthermore, if we refer to the distance between the swarm and the beacon, the swarm is not closing on the beacon. As depicted in figure 9(c) and figure 9(d),

the swarms only reach around 10 to 15 *cm* from the beacon when they move around the failed robots. This trend again continues for 5 failing robots in the environment.

3.3. Experimental Findings

So far, we have studied the effect of single nearest charger and shared nearest charger algorithms in an attempt to resolve the potential anchoring issues for power failure in the context of swarm beacon taxis. Based on these results, we observe issues with swarm taxis in line with [2] and with simple repair strategies are unable to mitigate these issues. With a *single nearest charger algorithm*, only one robot needs to share its energy with a faulty robot. However, since only one robot is sharing the energy, it has to give a large proportion of the required energy to the faulty robot, resulting to the major reduction on its own energy. This issue is crucial, and potentially not scalable, when the number of faulty robots increases. In an attempt to resolve this issue, we proposed an enhancement by increasing the number of simultaneous chargers . Here, we proposed a *shared nearest charger algorithm*, with three charging robots for each failing robot in the environment. Compared with the *single nearest charger algorithm*, the robots will no longer suffer with major energy reduction. However, when too many robots try to reach the faulty robots, docking and navigation is clearly an issue as robots interfere with each other to recharge the failing robots. Thus, a design of the docking and signalling algorithms than can improve the navigation is clearly needed. Due to these issues, we propose an immune-inspired solution inspired by the process of granuloma formation in immune systems. In the remaining part of the paper, we first describe the process granuloma formation in section 4. Secondly, we explain the computational model of granuloma formation in accordance with the idea of immuno-engineering as proposed in [19] in section 4.1 and finally we describe the granuloma formation algorithm in section 4.2, followed by an empirical investigation into the algorithms performance.

4. Granuloma Formation

An immune system is a complex system, involved in the defence of a host that has evolved in animals and plants over millions of years. There are many threats to hosts, and the immune system has evolved a variety of mechanisms to cope with these threats [18]. Of particular interest to our work is the formation of structures known as *granuloma* in response to certain pathogenic infection. Granulomas form in response to cells being infected by pathogens (in particular in response to infection by tuberculosis and Leishmanianas), and are structures that form of particular immune cells, known as T-cells that try and contain the infection in the cell and prevent cell to cell transmission of the pathogen [14]. The main cells involved in granuloma formation are; macrophages, T-cells and cytokines. [14] outline three main stages of granuloma formation:

1. T-cells are primed by antigen presenting cells;

2. Cytokines and chemokines are released by macrophages, activated T-cells and dendritic cells. The released cytokines and chemokines attract and retain specific cell populations.
3. The stable and dynamic accumulation of cells and the formation of the organised structure of the granuloma.
   These do vary slightly depending on location of the formation, be that in the liver or lung, for example, but the principles are the same.

In granuloma formation a cell called a macrophage will *'eat'* or engulf the pathogen in an attempt to prevent the pathogen from spreading. However, the pathogen may infect the macrophage and begin to duplicate. Therefore, despite the fact that macrophages are able to stop the infection, pathogens will use macrophages as a *'taxi'* to spread diseases leading to the cell lysis (the breaking down the structure of the cell). Infected macrophages then will emit a signal which indicates that they have been infected and this signal will lead other macrophages to move to the site of infection, to encase the infected cells, in effect then isolating the infected cells from the uninfected cells. This will ultimately lead to the formation of a granuloma structure, and in many cases the removal of the pathogenic material. By separating the infected macrophages, the robustness of the system will be maintained and the failure of one or more cells will not affect other cells and the overall task of the system can be maintained.

During granuloma formation, macrophages form a walling around the chronically infected macrophages as well as the bacteria leading to the formation of granuloma with the objective of separating the infected and uninfected cells as well as separating the bacteria from the healthy cells. By separating the infected macrophages, the robustness of the systems will be maintained and the failure of one or more cell will not affected other cell and the overall desired task of the systems can be achieved. Based on our literature survey in Section 3.1, macrophages have essentially four states:

- uninfected: able to take up bacteria (phagocyte) and if not activated quickly, it will become infected; not secreting anything
- infected: can still phagocytose and kill but function will decreases with increasing bacterial load, able to secrete chemokines
- activated: efficient at killing their intracellular bacterial load. In order to be activated, macrophages need to be activated by T-cell
- chronically infected: cannot phagocyte or kill; secrete chemokines

Figure 11, shows two phases of macrophages in granuloma formation, the uninfected and infected. In a), uninfected macrophages moves randomly until one of them detects the existance of bacteria (in b) where it will become infected macrophages and start to propagate a signal (in c), using chemokines. This signal will lead other macrophages to move towards the site of infections and form a granuloma to isolate the bacteria from other cell, which is in d.

For our purpose, the most important property in granuloma formation is the communication between macrophages which is determined by the level of chemokine secretion. Chemokines will not only attract other macrophages to move towards the site of infection but it will activate T-cells that will secrete cytokines to act as a signal for activation of macrophages. T-cells and activated macrophages are able to kill extra cellular bacteria that will control infections in immune systems. This is show in Figure 12. In this figure, we illustrate the different stages of macrophages and shows how the secretion of chemokines attract other macrophages. The infected macrophage which is in blue will secrete chemokines to attract other uninfected macrophages to the site of infection so that it can be isolated from the other cells and thus prevent further infection.

It is this early stage formation that we find interesting in the case of the swarm beacon taxis failures. We propose that there is a natural analogy between the potential repair of a swarm of robots, as in the situation of swarm beacon taxis, and the formation of a granuloma and removal of pathogenic material. We now explore this analogy in the context of immuno-engineering [19].

4.1. Computational Model of Granuloma Formation

In understanding the problem domain and the immune systems under study, we follow the Immuno-engineering approach that is proposed by [18]. Immuno-engineering is defined in [18] as:

> *'The abstraction of immuno-ecological and immuno-informatics principles, and their adaptation and application to engineered artefacts (comprising hardware and software), so as to provide these artefacts with properties analogous to those provided to organisms by their natural immune systems. '*

Immuno-engineering involves the adoption of conceptual framework [15] in AIS algorithm development and the problem-oriented perspective described in [4]. [18] propose a bottom-up approach to the engineering of such systems which will be achieved via an interdisciplinary approach which cuts across immunology, mathematics, computer science and engineering. Having explained the problem domain in which we are working, we now are able explore the biological background to our solution and take that forward towards a novel immune inspired aggregation algorithm for swarm robotic systems.

Based on the immunology surrounding the idea of granuloma formation, when studied in the context of the conceptual framework [15], we prepared an agent based simulation to provide the framework to allow us to understand how the process of granuloma formation occurs in the immune system that can provide insight into its behavioural aspects leading to an exploitation of biological characteristics in granuloma formation [7]. Details of the model are not included in this paper. By modelling and simulating the properties of granuloma formation we can attempt to formalise principles that govern the behaviour of cells in the system and apply them towards the development of a solution in our specific engineering problem. Through the analysis of our developed model and

simulation, we have constructed four design principles of self-healing in swarm robotic systems. The four principles for our algorithm development are:

1. The communication between agents in the system is indirect consisting number of signals to facilitate coordination of agents.
2. Agents in the systems react to defined failure modes by means of faulty from non-faulty using self-organising manner
3. Agents must be able to learn and adapt by changing their role dynamically.
4. Agents can initiate a self-healing process dependant to their ability and location.

As we have discussed in [20] applying the granuloma formation concepts to swarm robotics allows us to perform two related tasks. First, we may isolate faulty robots. Given that we can detect faulty robots, such robots may effect the performance of other robots and act as an 'anchor' to other robots, as we have shown above. In [20] we discuss how we can assume that certain signals can be sent by the robot which other functional robots nearby can recognise. These functional robots are then attracted towards the faulty robot, akin to how T-cells are attracted by cytokines emitted by an infected macrophage. A limited number of these robots then isolate the fault robot, akin to T-cells surrounding an infected macrophage, but still move around the faulty robot so that other functional robots in the swarm are no longer drawn to the 'anchor' point that could be the faulty robot. We now move to the development of a *granuloma inspired* approach to a self-healing swarm.

4.2. Granuloma Formation Algorithm

Using our knowledge gained from the development of a computational model, and the subsequent derivation of the design principles outlined above, we have derived the granuloma formation algorithm for solving the *anchoring issue* in swarm beacon taxis. We illustrate the process of granuloma formation algorithm in figure 13 which then is outlined in algorithm 5. The key differences between the granuloma formation algorithm and the shared charger algorithm is the number of functional robots that will surround and share the energy with the faulty robots varies, it is not pre-defined. This will be explained later.

As outlined in algorithm 5, in granuloma formation, failing robots will send signals that can be recognised by other functional robots. These functional robots are then attracted towards the faulty robot, akin to how T-cells are attracted by cytokines emitted by an infected macrophage in granuloma formation. A limited number of these robots then isolate the faulty robot, akin to T-cells surrounding an infected macrophage. The other robots which are not involved in isolating the faulty robot will ignore the failing and surrounding robots and treat them as if they were obstacles in a manner similar to the standard ω-algorithm.

In our proposed granuloma formation algorithm, the number of functional robots that will surround the faulty robots varies, it is not pre-defined. The number of robots required will be determined by the amount of energy required

begin
Deployment: robots are deployed in the environment
repeat
Random movement of the robot in the environment
Signal propagation: Faulty robots will emit faulty signal, where the average radius of the target neighbouring robot is R
Protection and rescue: Healthy robots will decide how many will perform protection and rescue according to algorithm 6
Repair: Sharing of energy between faulty and healthy robots according to algorithm 7
until *forever*
end

Algorithm 5: Overview of Granuloma Formation Algorithm

to repair the failing robot, together with the location of the faulty robot. Each faulty robot is able to evaluate their own energy level and position, and propagate that information to other robots, as outlined in algorithm 6. An illustration for this process is shown in figure 14. From this figure, when there exists a faulty robot(s), it will emit a signal to robots at a certain predefined radius (R). The robots that receive the signal will communicate, exchanging the information on their locations and their current energy. This is according to the energy transfer rules explain in algorithm 7, where the nearest functional robot will come alongside the robot, and share an amount of energy. If the energy that can be donated by the nearest functional robot is enough the faulty robots will stop sending the information to the other robots. However, if the energy is not enough, the faulty robot will continue to request energy from other functional robots in the environment. This will be repeated until enough energy information is obtained and then the isolating and sharing of energy will then take place.

begin
Evaluate $egy_{needed}(t)$
Send $egy_{needed}(t)$ *to peers within* R
Receive $egy_{peer}(t-x)$ *from peers within* R
forall $egy_{peer}(t-x)$ *received* **do**
if $egy_{peer}(t-x)$ *received* $< egy_{needed}$ **then**
Send $egy_{needed}(t)$ *to peers within* R
else
Stop sending $egy_{needed}(t)$ *to peers within* R
end
end
end

Algorithm 6: Algorithm for containment and repair according to energy and position of robots

The basic terms used in the algorithm are as follow:

```
begin
    Evaluate egy_self(t) and pos_self(t)
    Send egy_self(t) and pos_self(t) to peers within R
    Receive egy_peer(t − x) and pos_peer(t − x) from peers
    forall egy_peer(t − x) received do
        if egy_peer(t − x) < egy_min then
            R = match egy_peer(t − x), egy_self(t)
            Store egy_peer(t − x) in inbound queue
        else
            R' = not match egy_peer(t − x), egy_self(t)
            Add egy_peer(t − x) to outbound queue
        end
    end
    forall egy_peer(t − x) in inbound queue do
        Add signature
        Store egy_peer(t − x) in robot list
        forall egy_peer(t − x) in robot list do
            if egy_self(t) < egy_threshold then
                Evaluate pos_peer(t − x)
                Sort pos_peer(t − x)in ascending order
                Move to nearest pos_peer(t − x)
            end
        end
    end
    forall egy_peer(t − x) in outbound queue do
        Delete signature
    end
end
```

Algorithm 7: Algorithm for containment and repair according to energy and position of robots

- $pos_{self}(t)$: position of self robot
- $pos_{peer}(t-x)$: position of peer robots
- $egy_{self}(t)$: energy of self robot
- $egy_{peer}(t-x)$: energy of peer robots
- egy_{needed}: energy needed by failing robot
- $egy_{threshold}$: the limit of the energy that is needed

In the real robot scenario, during the repairing phase, donor robots will share their energy with the faulty robots. During this time, we will adopt the ideas of signalling mechansim in granuloma formation that is explain in section 4. In granuloma formation, different signals are emitted by macrophages to inform other cells their current status. In granuloma formation algorithm, we will use different color of LED signals during the repairing phase. The green LED signals will be emitted by the faulty and donor robots that are involved in this phase notifying the other robots to ignore them and continue moving towards the beacon. This is illustrated in figure 15.

4.3. Experimental Investigation to the Granuloma Formation Algorithm

For this experiment, we follow the experimental protocol that is described in Section 3.1. We cast our hypotheses as follows:

$H6_0$: *The use of a granuloma formation algorithm (M4) does not improve the ability of the robots in the system to reach the beacon when compared to M2 when more than two faulty robots are introduced to the systems*

$H7_0$: *The use of a granuloma formation algorithm (M4) does not improve the ability of the robots in the system to reach the beacon when compared to M3 when more than two faulty robots are introduced to the systems*

This experiment tests hypothesis $H6_0$ and $H7_0$ to determine the performance of the granuloma formation algorithm. The first set of experiments are on swarm beacon taxis with R_{SF} and R_{TF}. We than continue to test the algorithm with R_{MF}. M4 is assessed with the same scenario as the single nearest charger algorithm (M2), shared charger algorithm (M3) and ω-algorithm (M1) as the base line experiments. We than compare the algorithms and examine at the statistical and scientific differences with ranksum and A measure tests. Results for M4 are illustrated in figure 16(a), 16(b), 16(c) and 16(d). Based on the graphs, swarm beacon taxis with M4 is able to reach the beacon with R_{SF}, R_{TF} and R_{MF}.

We now measure the significant difference between M4 with single nearest charger algorithm (M2) and shared charger algorithm (M3). We also compare M4 with ω-algorithm (M1). All results are shown in table 7 and table 8. With R_{SF}, R_{TF}, there is no significant difference between M4, M3 and M2. This explains that having a small number of failing robots in the environment, most

Table 7: P and A value for mean centroid distance on robots between M4 and M3 in 1000 simulation seconds.

M4/M2	**ranksum P**	**A measure**
One fail	0.7903	0.4600
Two fails	0.4722	0.4000
Three fails	0.0350	0.8900
Four fails	1.0650e-04	1
Five fails	1.7861e-04	1

Table 8: P and A value for mean centroid distance on robots between M4 and M2 in 1000 simulation seconds.

M4/M3	**ranksum P**	**A measure**
One fail	0.2563	0.3450
Two fails	0.0307	0.2100
Three fails	0.0081	0.8550
Four fails	1.0650e-04	1
Five fails	1.7856e-04	1

of the mechanisms is able to make the systems recover and operate to perform the needed task. The crucial part is when there are three or more robots that fail. The algorithm must cater for this type of failure to achieve robustness. If we compare M4 with M3 and M2 with R_{MF}, it shows that M4 performs significantly better than M3 and M2 with 3, 4 and 5 failing robots. M4 also performs better than the baseline experiment in this paper.

Form these experiments, even though M4 does not differ significantly with M2 and M3 with less than one and two failing robots, with three, four and five failing robots, the experiments reject the hypothesis $H6_0$ at the default of α= 0.05 significance level, which is indicated by the p value = 0.0350, p value = 1.0650e-04 and p value = 1.7861e-04 that has fallen below the α. The 95% confidence interval on the mean centroid distance of robots to the beacon is less than 5 *cm* is obtained from this experiment. The experiments also reject the hypothesis $H7_0$ at the default of α= 0.05 significance level with more than two failing robots are introduced to the system, which is indicated by the p value = 0.0081, p value = 1.0650e-04 and p value = 1.7856e-04 that has fallen below the α. The 95% confidence interval on the mean centroid distance of robots to the beacon is less than 5 *cm* is obtained from this experiment.

As in previous experiments, all robots start with equal energy which is 5000 Joules. Figure 17(a), 17(b) and 17(c) show the energy of 10 robots during 1000 simulation seconds with different failing robots in the environment with granuloma formation algorithm. Having three to five failing robots, we can see that the average energy level of all robots in the environment are still above the minimum energy level which is 500 Joules. This explain that the robots can still share their energy even though half of the robots are suffering with low energy

levels. The swarm is able to continue operating even with such failures.

5. Conclusion

Our work has shown the use of single nearest charger algorithm, shared charger algorithm and granuloma formation algorithm for self-healing mechanism in swarm robotic systems applied in a case study of swarm beacon taxis. We have showed that both single nearest charger algorithm and shared nearest charger algorithm work equally well with a low number of faulty robots in the systems. From the statistical analysis, we have shown that single and shared charging algorithms work equally well with at most three faulty robots in the system. Based on the results obtained from both algorithms, we propose a new *granuloma formation algorithm* inspired by the process of granuloma formation in immune systems. The experiments show that the granuloma formation algorithm works well even when half of the robots in the swarm are experiencing low energy levels. In the future, we will eventually implement it on physical e-puck robots [10]. This will help us on studying the applicability of the algorithm to real robots environment.

References

References

[1] Bayindir L, Şahin E (2007) A review of studies in swarm robotics. Turkish Journal of Electrical Engineering and Computer Science 15(2):115–147

[2] Bjerknes J, Winfield A (2010) On fault-tolerance and scalability of swarm robotic systems. In: Proc. Distributed Autonomous Robotic Systems (DARS)

[3] Bjerknes JD (2009) Scaling and fault tolerance in self-organized swarms of mobile robots. PhD thesis, University of the West of England, Bristol, UK.

[4] Freitas A, Timmis J (2007) Revisiting the foundations of artificial immune systems for data mining. IEEE Transaction on Evolutionary Computation 11(4):521–540

[5] Gerkey B, Vaughan RT, Howard A (2003) The player/stage project: Tools for multi-robot and distributed sensor systems. In: Proceedings of the 11th International Conference on Advanced Robotics (ICAR 2003), pp 317–323

[6] Haek MA, Ismail AR, Basalib AOA, Makarim N (2015) Exploring energy charging problem in swarm robotic systems using foraging simulation. Jurnal Teknologi 76 (1):239–244

[7] Ismail AR, Timmis J (2010) Towards self-healing swarm robotic systems inspired by granuloma formation. In: Special Session: Complex Systems Modelling and Simulation, part of ICECCS 2010, IEEE, pp 313 – 314

[8] Melhuish C, Hoddell OHS (1998) Collective sorting and segregation in robots with minimal sensing. In: Proceedings of the fifth international conference on simulation of adaptive behavior on From animals to animats 5, MIT Press, pp 465–470

[9] Melhuish C, Kubo M (2007) Collective energy distribution: Maintaining the energy balance in distributed autonomous robots using trophallaxis. In: Alami R, Chatila R, Asama H (eds) Distributed Autonomous Robotic Systems 6, Springer Japan, pp 275–284

[10] Mondada F, Bonani M, Raemy X, Pugh J, Cianci C, Klaptocz A, Magnenat S, Zufferey JC, Floreano D, Martinoli A (2009) The e-puck, a Robot Designed for Education in Engineering. In: Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions, vol 1, pp 59–65

[11] Nembrini J, Winfield A, Melhuish C (2002) Minimalist coherent swarming of wireless networked autonomous mobile robots. In: Proceedings of the seventh international conference on simulation of adaptive behavior: From animals to animats (SAB '02), MIT Press, vol 7, pp 373 – 382

[12] Ohkura K, Yu T, Yasuda T, Matsumura Y, Goka M (2015) Robust swarm robotics system using cma-neuroes with incremental evolution. International Journal of Swarm Intelligence and Evolutionary Computation 4 (125):1–10

[13] Şahin E (2005) Swarm robotics: From sources of inspiration to domains of application. In: International Workshop of Swarm Robotics (WS 2004), Springer, LNCS, vol 43, pp 10–20

[14] Sneller MCM (2002) Granuloma formation, implications for the pathogenesis of vasculitis. Cleveland Clinic Journal of Medicine 69(2):40–43

[15] Stepney S, Smith R, Timmis J, Tyrell A, Neal MJ, Hone ANW (2005) Conceptual framework for artificial immune systems. Unconventional Computing 1(3):315–338

[16] Støy K (2001) Using situated communication in distributed autonomous mobile robotics. In: In Proceedings of the 7th Scandinavian conference on artificial intelligence, Amsterdam: IOS, pp 44–52

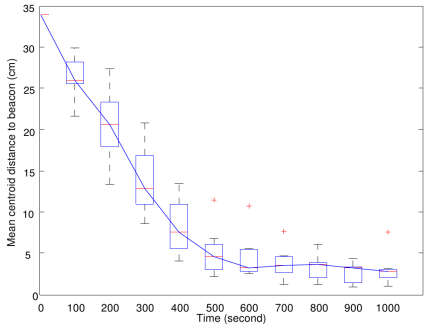
[17] Støy K, Shen WM, Will P (2002) Global locomotion from local interaction in self-reconfigurable robots. In: 7th International Conference on Intelligent and Autonomous Systems (IAS7), pp 309–316

[18] Timmis J, Andrews P, Owens N, Clark E (2008) An interdisciplinary perspective on artificial immune system. Evolutionary Intelligence 1(1):5–26

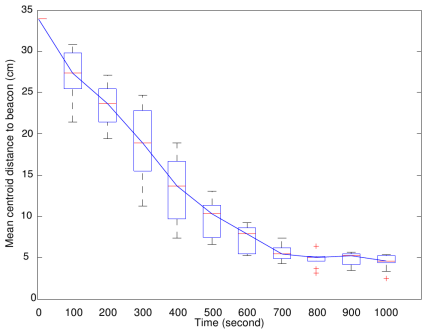
[19] Timmis J, Hart E, Hone A, Neal M, Robins A, Stepney S, Tyrrell A (2008) Immuno-engineering. In: 2nd International Conference on Biologically Inspired Collaborative Computing, IFIP, vol 268, pp 3–17

[20] Timmis J, Tyrrell A, Mokhtar M, Ismail AR, Owens N, Bi R (2010) An artificial immune system for robot organisms. In: Levi P, Kernback S (eds) Symbiotic Multi-Robot Organisms: Reliability, Adaptability and Evolution, vol 7, Springer, pp 279–302

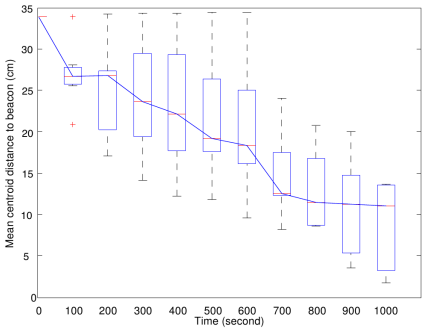
[21] Vargha A, Delaney HD (2000) Critique and improvement of cl common language effect size statistics of mcgraw and wong. Journal of Educational and Behavioral Statistics 25 (2):101–132



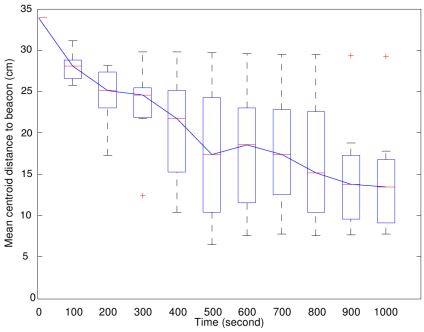
(a)



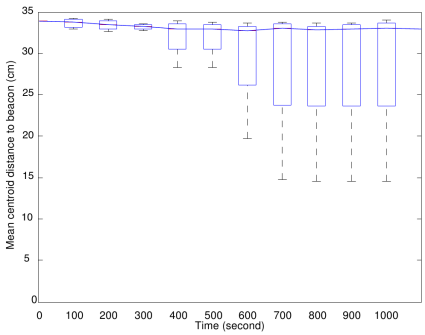
(b)



(c)

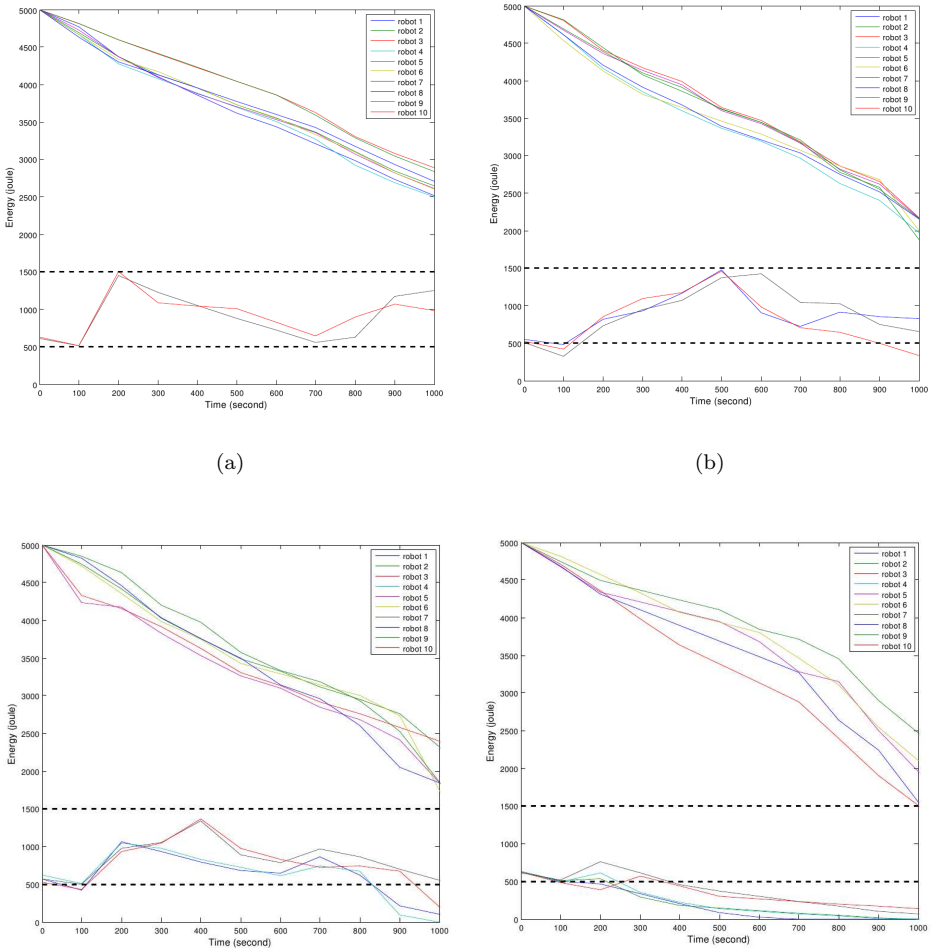


(d)



(e)

Figure 9: Boxplots of the distance between swarm centroid and beacon as a function of time for 10 experiments using shared charger algorithm with one faulty robot (9(a)), two failing robots simultaneously at t=100 seconds (9(b)), three failing robots simultaneously at t=100 seconds(9(c)), four failing robots simultaneously at t=100 seconds (9(d)) and five failing robots simultaneously at t=100 seconds (9(e)). The centre line of the box is the median while the upper edge of the box is the 3^{rd} quartile and the lower edge of the box is the 1^{st} quartile. The swarm reach the beacon at t=650 for one faulty robot, at t=800 for two failing robots. The swarm does not reach the beacon with three, four and five failing robots.



(a) (b)

(c) (d)

Figure 10: The energy for swarm of ten robots using shared nearest charger algorithm with two (10(a)), three (10(b)), four (10(c)) and five (10(d)) failing robots.

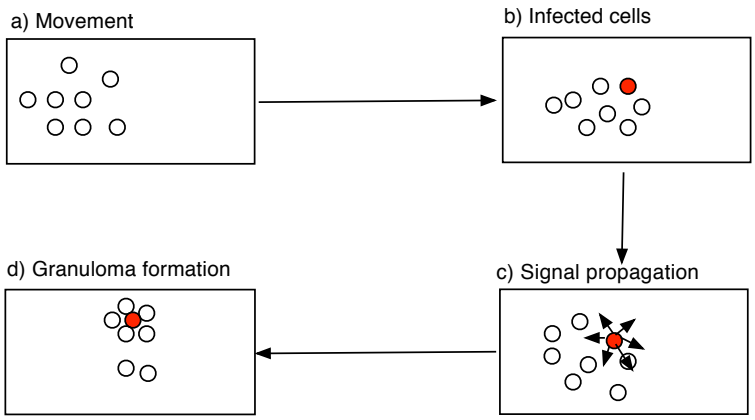


Figure 11: Simple scenario: formation of granuloma

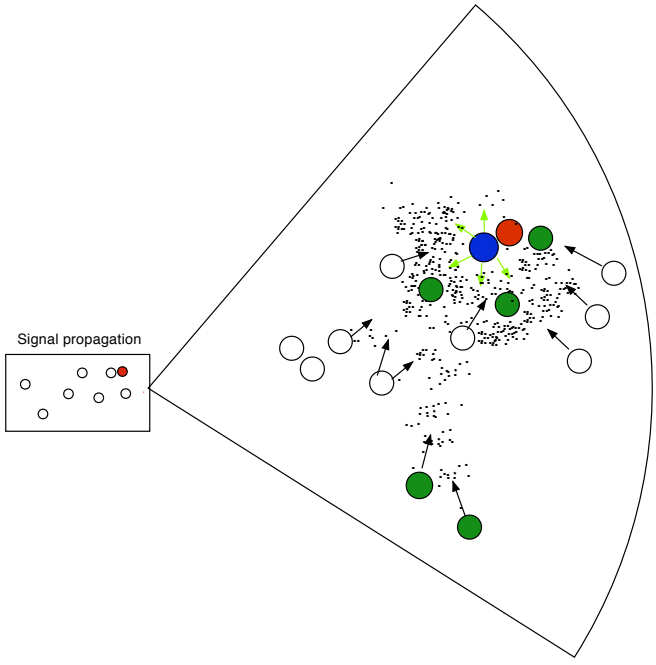


Figure 12: Chemokines propagation in granuloma formation. Legend of colours: red, bacteria; blue, infected macrophages; white, uninfected; green, T-cell

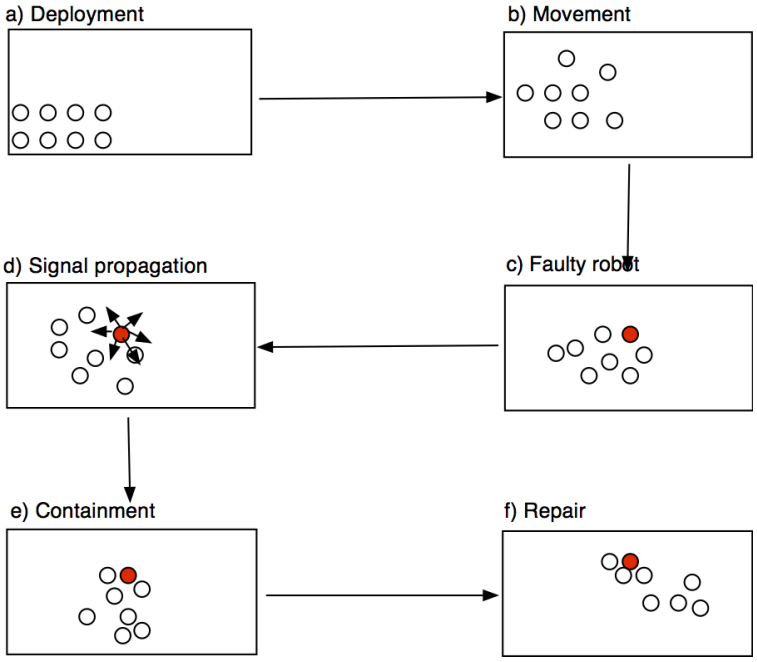


Figure 13: Simple Scenario: granuloma formation algorithm

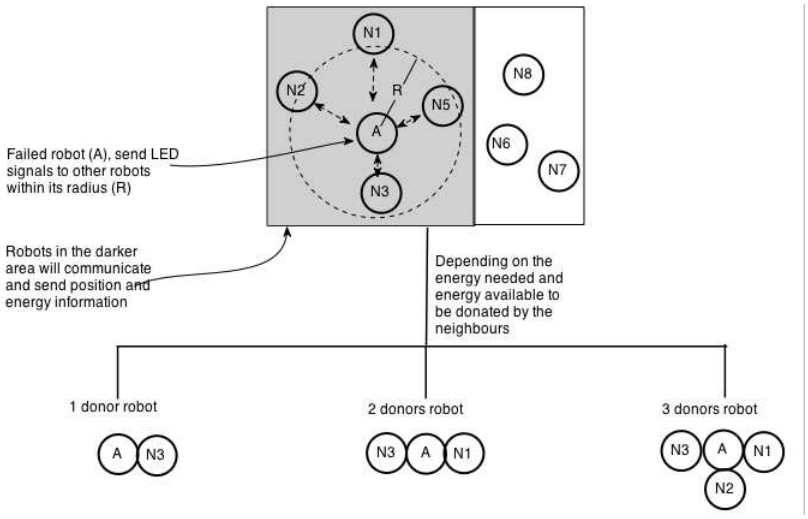


Figure 14: The number of functional robots that will share their energy will be based on its position, their current energy and the energy needed from the faulty robot.

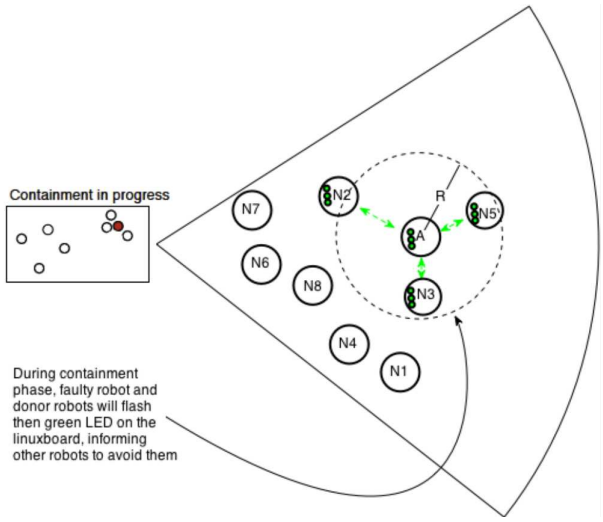
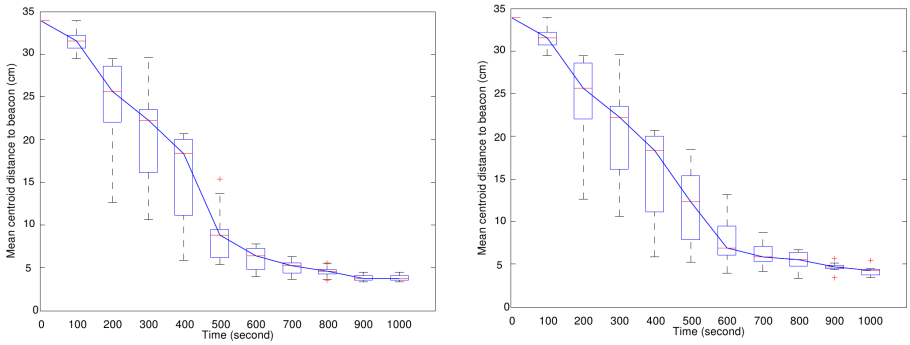
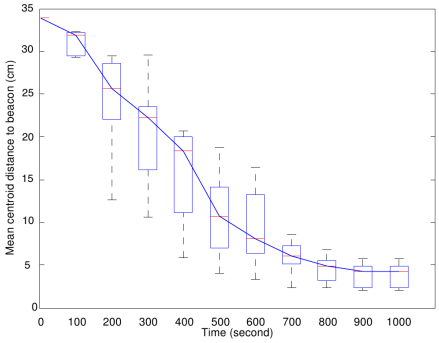


Figure 15: The repairing phase of granuloma formation algorithm. Robot that involve in this phase will emit the green LED signals notifying the other robots that they need to be ignored

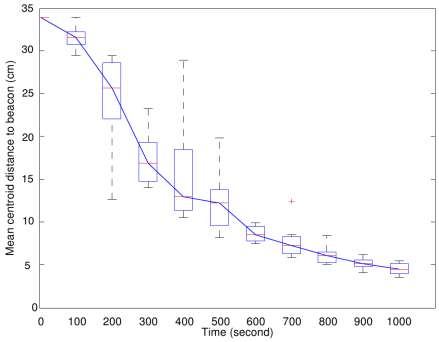


(a)

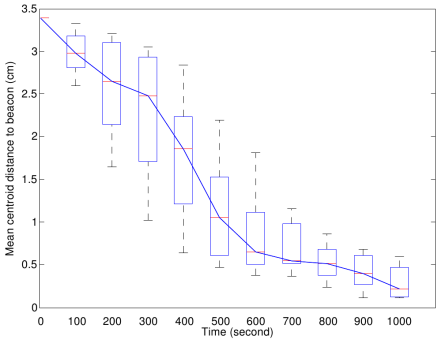
(b)



(c)

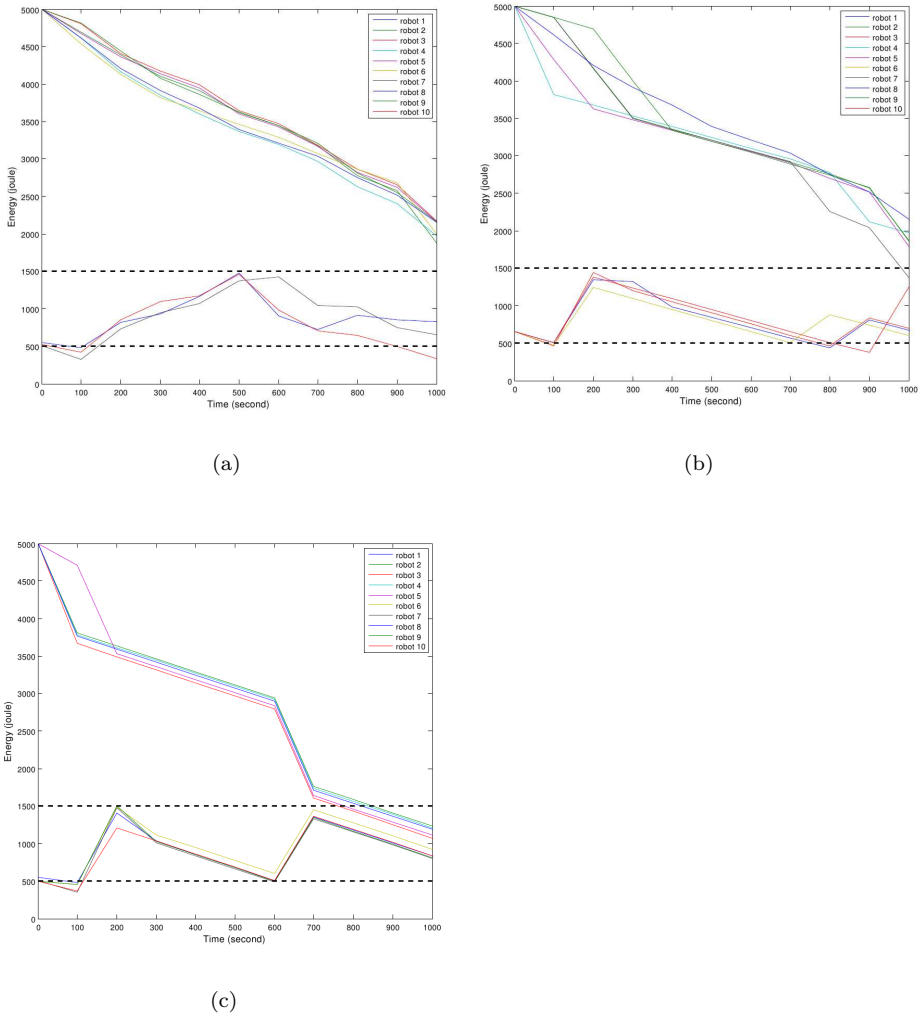


(d)



(e)

Figure 16: Boxplots of the distance between swarm centroid and beacon as a function of time for 10 experiments using granuloma formation algorithm with one faulty robot (16(a)), two failing robots (16(b)), three failing robots (16(c)), four failing robots (16(d)) and five failing robots (16(e)) simultenously at t=100 seconds for $H6_0$ and $H7_0$. The centre line of the box is the median while the upper edge of the box is the 3^{rd} quartile and the lower edge of the box is the 1^{st} quartile. The swarm reach the beacon at t=650 for one faulty robot, at t=800 for two failing robots, at t=850 for three failing robots and t=950 to t=1000 for four and five failing robots.



(a)

(b)

(c)

Figure 17: The energy autonomy for swarm of 10 robots using shared nearest charger algorithm with two (10(a)), three (10(b)), four (10(c)) and five (10(d)) failing robots.